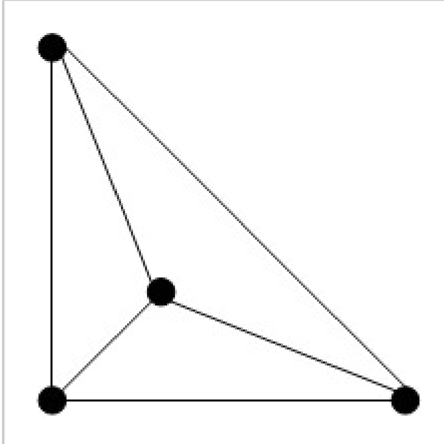
Teoria dos grafos

Origem: Wikipédia, a enciclopédia livre.

A **teoria dos grafos** é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto. Para tal são empregadas estruturas chamadas de grafos, G(V,E), onde V é um conjunto não vazio de objetos denominados vértices e E é um subconjunto de pares não ordenados de V, chamados arestas.



Grafo com quatro vértices e 6 arestas. É um grafo completo, conexo e planar.

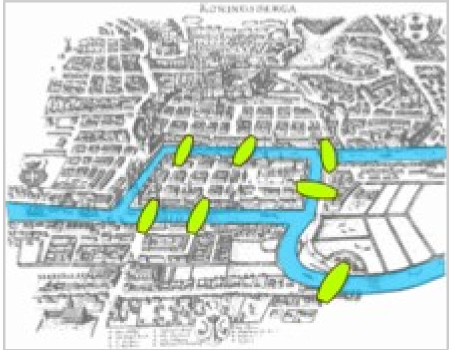
Dependendo da aplicação, arestas podem ou não ter direção, pode ser permitido ou não arestas ligarem um vértice a ele próprio e vértices e/ou arestas podem ter um peso (numérico) associado. Se as arestas têm uma direção associada (indicada por uma seta na representação gráfica) temos um *dígrafo* (grafo orientado). Um grafo com um único vértice e sem arestas é conhecido como **grafo trivial**.

Estruturas que podem ser representadas por grafos estão em toda parte e muitos problemas de interesse prático podem ser formulados como questões sobre certos grafos. Por exemplo, a estrutura de ligações da Wikipédia pode ser representada por um dígrafo: os vértices são os artigos da Wikipédia e existe uma aresta do artigo *A* para o artigo *B* se e somente se *A* contém um link para *B*. Dígrafos são também usados para representar máquinas de estado finito. O desenvolvimento de algoritmos para manipular grafos é um tema importante da ciência da computação.

Índice

Histórico

O artigo de Leonhard Euler, publicado em 1736, sobre o problema das *sete pontes de Königsberg*, é considerado o primeiro resultado da teoria dos grafos.[1] É também considerado um dos primeiros resultados topológicos na geometria; isto é, não dependente de quaisquer medidas. Isso ilustra a profunda conexão entre a teoria dos grafos e topologia.


O problema das pontes de Königsberg

Mais de um século após a publicação do artigo de Euler sobre as pontes de Königsberg e enquanto Listing introduzia o conceito de topologia, Cayley foi levado por um interesse em formas analíticas particulares decorrentes do cálculo diferencial para estudar uma classe particular de grafos, as *árvores*[2]. Esse estudo teve diversas implicações na química teórica. As técnicas usadas por ele eram relacionadas a enumeração de grafos com propriedades particulares.

O primeiro livro didático sobre teoria dos grafos foi escrito por Dénes König e publicado em 1936[3].

Um dos problemas mais conhecidos em teoria dos grafos é problema das quatro cores: "É possível que qualquer mapa desenhado num plano, dividido em regiões, possa ser colorido com apenas quatro cores de tal forma que as regiões vizinhas não partilhem a mesma cor?". O primeiro a notar o problema das quatro cores foi August Ferdinand Möbius em 1840. Em 1852, Francis Guthrie escreveu em uma carta para seu irmão Frederick, estudante na University College London, sobre o problema. Mas nenhum deles conseguiu resovê-lo. Então Frederick perguntou a um de seus professores, DeMorgan[4].

Definições de grafos e dígrafos

Na literatura, as definições básicas da teoria dos grafos variam bastante. Aqui estão as convenções usadas nesta enciclopédia.

Um **grafo direcionado** (também chamado **dígrafo** ou **quiver**) consiste de

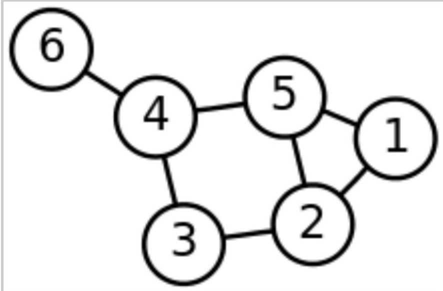
- um conjunto *V* de *vértices*,
- um conjunto *E* de *arestas* e
- mapas $s, t : E \to V$, onde $s(e)$ é a *fonte* e $t(e)$ é o *alvo* da aresta direcionada *e*.

Um **grafo não direcionado** (ou simplesmente **grafo**) é dado por

- um conjunto *V* de *vértices*,
- um conjunto *E* de *arestas* e
- uma função $w : E \to \mathrm{P}(V)$ que associa a cada aresta um subconjunto de dois ou de um elemento de *V*, interpretado como os pontos terminais da aresta.

Em um grafo ou dígrafo **com pesos**, uma função adicional E → **R** associa um valor a cada aresta, o que pode ser considerado seu "custo"; tais grafos surgem em problemas de rota ótima tais como o problema do caixeiro viajante.

Representação gráfica



Um grafo com seis vértices e 7 arestas

Os grafos são geralmente representados graficamente da seguinte maneira: é desenhado um círculo para cada vértice, e para cada aresta é desenhado um arco conectando suas extremidades. Se o grafo for direcionado, seu sentido é indicado na aresta por uma seta.

O grafo de exemplo exibido à direita é um grafo simples com o conjunto de vértices $V = \{1, 2, 3, 4, 5, 6\}$ e um conjunto de arestas $E = \{ \{1,2\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,4\}, \{4,5\}, \{4,6\} \}$ (com o mapeamento *w* sendo a identidade).

Note que essa representação gráfica não deve ser confundida com o grafo em si (a estrutura abstrata, não-gráfica). Diferentes representações gráficas podem corresponder ao mesmo grafo.[5] O que importa é quais vértices estão conectados entre si por quantas arestas.

O trabalho pioneiro de William Thomas Tutte foi de grande influência no desenho de grafos. Entre outras realizações, ele introduziu o uso de técnicas da álgebra linear para desenhar grafos.

Convenções alternativas para a representação de grafos incluem representações por adjacência como empacotamento de círculos, onde vértices são representados como regiões disjuntas no plano e arestas são representadas por adjacências entre regiões; representações por intersecção no qual vértices são representados como objetos geométricos não disjuntos e arestas são representadas por suas intersecções; representações por visibilidade onde vértices são representados por regiões no plano e arestas são representadas por regiões com uma linha de visão desobstruída para cada vértice; desenhos confluentes, nos quais arestas são curvas suaves dentro de trilhos de trem; texturas onde vértices são linhas horizontais e arestas são linhas verticais[6]; e visualizações da matriz de adjacência de um grafo.

O desenho de grafos em superfícies diferentes do plano é também estudado.

Armazenamento de grafos em memória

Há diversas maneiras de armazenarmos grafos em computadores. A estrutura de dados usada dependerá tanto da estrutura do grafo quanto do algoritmo usado para manipulá-lo. Teoricamente, podemos dividir entre estruturas do tipo lista e do tipo matriz, mas em aplicações reais, a melhor estrutura é uma combinação de ambas. Estruturas do tipo lista são frequentemente usadas em grafos esparsos (grafos que possuem um pequeno número de arestas em relação ao número de vértices, oposto ao grafo denso, onde o número de arestas se aproxima do máximo de arestas possível) já que exigem menor uso da memória. Por outro lado, estruturas do tipo matriz fornecem um rápido acesso em algumas aplicações, mas podem consumir uma grande quantidade de memória.

Estruturas do tipo lista incluem a lista de adjacência que associa a cada vértice do grafo uma lista de todos os outros vértices com os quais ele tem uma aresta e a lista de incidência, que armazena para cada vértice uma lista de objetos que representam as arestas incidentes a esse vértice[7][8].

Estruturas do tipo matriz incluem a matriz de incidência, uma matriz de 0's e 1's com suas linhas representando vértices e suas colunas as arestas e a matriz de adjacência onde ambas linhas e colunas possuem vértices. Em ambos casos um 1 indica dois objetos adjacentes e 0 indica dois objetos não adjacentes.

Definições de teoria dos grafos

Relações de incidência e de adjacência

Se uma aresta conecta dois vértices, então esses dois vértices são ditos **incidentes** à aresta. Usando o grafo ao lado como exemplo temos: 1 é incidente a 2 e 5, mas não é incidente a 3, 4 ou 6; 4 é incidente a 5, 3 e 6, mas não a 2 nem a 1.

Dois vértices são considerados **adjacentes** se uma aresta existe entre eles. No grafo acima, os vértices 1 e 2 são adjacentes, mas os vértices 2 e 4 não são. O conjunto de **vizinhos** de um vértice consiste de todos os vértices adjacentes a ele. No grafo-exemplo, o vértice 1 possui 2 vizinhos: vértice 2 e vértice 5.

Na computação, um grafo finito direcionado ou não-direcionado (com, digamos, n vértices) é geralmente representado por sua **matriz de adjacência**: uma matriz n-por-n cujo valor na linha i e coluna j fornece o número de arestas do i-ésimo ao j-ésimo vértices.

Valência (Grau)

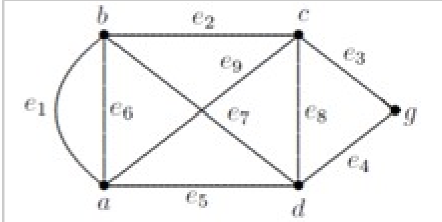
A **valência** (ou **grau**) de um vértice é o número de arestas incidentes a ele. Se houver laços, serão contados duas vezes. No grafo de exemplo os vértices 1 e 3 possuem uma valência de 2, os vértices 2, 4 e 5 têm a valência de 3 e o vértice 6 tem a valência de 1. Se E é finito, então a valência total dos vértices é o dobro do número de arestas. Em um dígrafo, distingue-se o **grau de saída** (o número de arestas saindo de um vértice) e o **grau de entrada** (o número de arestas entrando em um vértice). O grau de um vértice em um dígrafo é igual à soma dos graus de saída e de entrada. O grau de um vértice é definido somente quando o número de arestas incidentes sobre o vértice é finito.

- **Lema do aperto de mãos** diz que se os convidados de uma festa apertarem as mãos quando se encontrarem pela primeira vez, o número de convidados que apertam a mão um número ímpar de vezes é par. Também em grafos não direcionados a soma dos graus de todos os vértices é igual ao dobro do número de arestas.

Passeio

Um passeio p entre dois vértices A e B, definido como $p(A,B)$, é uma lista alternada de vértices e arestas $p(A,B) = v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ tal que

- $A = v_0$ e $B = v_k$
- existe no mínimo uma aresta
- para $1 \leq i \leq k$ a aresta e_i incide sobre v_{i-1} e v_i



Grafo com cinco vértices e 9 arestas

O tamanho de um passeio, definido por $|p|$ corresponde ao número de arestas que possui, incluindo as repetições[9]. Na figura ao lado, um passeio do vértice a até o vértice d possível é a lista $p(a,d) = a, e_9, c, e_2, b, e_1, a, e_9, c, e_8, d$ de tamanho 5. Note que a aresta e_9 se repete. É necessário listarmos as arestas em um passeio para distinguirmos entre as diversas arestas existentes quando estamos trabalhando em um grafo que não é simples. Em um grafo simples, basta listarmos os vértices[9].

Caminho

Caminho é uma sequência de vértices tal que de cada um dos vértices existe uma aresta para o vértice seguinte. Um caminho é chamado **simples** se nenhum dos vértices no caminho se repete. O **comprimento do caminho** é o número de arestas que o caminho usa, contando-se arestas múltiplas vezes. O **custo de um caminho** num grafo balanceado é a soma dos custos das arestas atravessadas. Dois caminhos são **independentes** se não tiverem nenhum vértice em comum, exceto o primeiro e o último.

No grafo de exemplo, (1, 2, 5, 1, 2, 3) é um caminho com comprimento 5, e (5, 2, 1) é um caminho simples de comprimento 2.

- **Caminho euleriano** em um grafo é o caminho que usa cada **aresta** exatamente uma vez. Se tal caminho existir, o grafo é chamado *traversável*. Um **ciclo euleriano** é um ciclo que usa cada aresta exatamente uma vez.
- **Caminho hamiltoniano** em um grafo é o caminho que visita cada **vértice** exatamente uma vez. Um **ciclo hamiltoniano** é um ciclo que visita cada vértice uma só vez. O grafo do exemplo contém um caminho hamiltoniano. Enquanto determinar se um dado grafo contém um caminho ou ciclo euleriano é trivial, o mesmo problema para caminhos e ciclos hamiltonianos é trabalhoso.

Ciclo (ou circuito) é um caminho que começa e acaba com o mesmo vértice. Ciclos de comprimento 1 são laços. No grafo de exemplo, (1, 2, 3, 4, 5, 2, 1) é um ciclo de comprimento 6. Um **ciclo simples** é um ciclo que tem um comprimento pelo menos de 3 e no qual o vértice inicial só aparece mais uma vez, como vértice final, e os outros vértices aparecem só uma vez. No grafo acima, (1, 5, 2, 1) é um ciclo simples. Um grafo chama-se **acíclico** se não contém ciclos simples.

Laço (*loop*) num grafo ou num dígrafo é uma aresta *e* em *E* cujas terminações estão no mesmo vértice.

Tipos de grafos

- **Grafo simples** é um grafo não direcionado, sem laços e existe no máximo uma aresta entre quaisquer dois vértices (sem arestas paralelas). Para um grafo simples, o número de vizinhos de um vértice é igual à sua valência.
- **Grafo completo** é o grafo simples em que, para cada vértice do grafo, existe uma aresta conectando este vértice a cada um dos demais. Ou seja, todos os vértices do grafo possuem mesmo grau. O grafo completo de *n* vértices é frequentemente denotado por K_n. Ele tem $n(n-1)/2$ arestas (correspondendo a todas as possíveis escolhas de pares de vértices).
- **Grafo nulo** é o grafo cujo conjunto de vértices é vazio.
- **Grafo vazio** é o grafo cujo conjunto de arestas é vazio.
- **Grafo trivial** é o grafo que possui apenas um vértice e nenhuma aresta.
- **Grafo regular** é um grafo em que todos os vértices tem o mesmo grau.
- **Multigrafo** é um grafo que permite múltiplas arestas ligando os mesmos vértices (arestas paralelas).
- **Pseudografo** é um grafo que contém arestas paralelas e laços.
- **Grafo conexo** um grafo é *conexo* se for possível estabelecer um caminho de qualquer vértice para qualquer outro vértice de um grafo. Se for sempre possível estabelecer um caminho de qualquer vértice para qualquer outro vértice mesmo depois de remover *k*-1 vértices, então diz-se que o grafo está ***k*-conexo**. Note que um grafo está *k*-conexo se, e somente se, contém *k* caminhos independentes entre qualquer par de vértices. O grafo de exemplo acima é conexo (e portanto 1-conexo), mas não é 2-conexo. Em um grafo genérico *G*, o **corte** associado a um conjunto *X* de vértices é o conjunto de todas as arestas que têm uma ponta em *X* e outra em $V(G) - X$, onde $V(G)$ é o conjunto de todos os vértices pertencentes ao grafo *G*.
- **Ponto de articulação** ou **Vértice de corte** é um vértice cuja remoção desliga um grafo. Uma **ponte** é uma aresta cuja remoção desliga um grafo. Um **componente biconectado** é um conjunto máximo de arestas tal que qualquer par de arestas do conjunto fazem parte de um ciclo simples comum. O **contorno** de um grafo é o comprimento do ciclo simples mais curto no grafo. O contorno de um grafo acíclico é, por definição, infinito.
- **Árvore** é um grafo simples acíclico e conexo. Às vezes, um vértice da árvore é distinto e chamado de *raiz*. As árvores são muito usadas como estruturas de dados em informática (veja estrutura de dados em árvore).
- **Floresta** é um conjunto de árvores; equivalentemente a uma floresta, em algum grafo acíclico.
- **Subgrafo** de um grafo *G* é um grafo cujo conjunto dos vértices é um subconjunto do conjunto de vértices *G*, cujo conjunto de arestas é um subconjunto do conjunto de arestas de *G*, e cuja função *w* é uma restrição da função de *G*
- **Subgrafo gerador** é aquele obtido pela remoção de uma ou mais arestas de um outro grafo, dizemos então que este novo grafo obtido é gerador do

primeiro,

- **Subgrafo induzido** é obtido pela remoção de vértices e consequente das arestas relacionadas com ele de um outro grafo, dizemos que este novo grafo é um grafo induzido do original.
- **Grafo parcial** de um grafo *G* é um subgrafo com o mesmo conjunto de vértices que *G*. Uma **árvore parcial** é um grafo parcial que é árvore. Todo grafo tem pelo menos uma árvore parcial.
- **Clique** em um grafo é um subgrafo que também é um grafo completo. No grafo do exemplo acima, os vértices 1, 2 e 5 formam um clique.
- **Conjunto independente** em um grafo é um conjunto de vértices não adjacentes entre si. No exemplo acima, os vértices 1, 3 e 6 formam um conjunto independente e 3, 5 e 6 são outro conjunto independente.
- **Grafo planar** é aquele que pode ser representado em um plano sem qualquer intersecção entre arestas. O grafo do exemplo é planar; o grafo completo de n vértices, para $n > 4$, não é planar.
- **Grafo bipartido** é o grafo cujos vértices podem ser divididos em dois conjuntos, nos quais não há arestas entre vértices de um mesmo conjunto. Para um grafo ser bipartido ele não pode conter circuitos de comprimento ímpar.
    1. **Se um grafo G é bipartido, todo o circuito de G possui comprimento par.**Sejam V1 e V2 os dois conjuntos em que, de acordo com a definição de grafo bipartido, se particiona V(G). Toda a aresta de G conecta um vértice em V1 com outro em V2. Assim sendo, se X for um vértice de V1, para "voltar" a esse vértice terá de se ir a V2 e voltar a V1 um número indeterminado de vezes, e de cada vez serão percorridas duas arestas, uma de um vértice em V1 para um vértice em V2 e outra de um vértice em V2 para um vértice em V1. Logo, o número de arestas a percorrer será par, ou seja, o comprimento do circuito é par.
    2. **Se todo o circuito de um grafo G possui comprimento par, então o grafo é bipartido.**Seja G um grafo em que todo o circuito tem comprimento par, e seja X um vértice de G. Denotemos por V1 o conjunto formado por X e por todos os vértices cuja distância a X é par. Seja V2 = V(G)\V1 (isto é, o conjunto formado pelos vértices de G que não pertencem a V1). Pretende mostrar-se que não existe qualquer aresta que conecte vértices de V1 ou vértices de V2. Suponhamos a existência de tal aresta, isto é, suponhamos a existência de dois vértices em V1 (ou V2), digamos Xi e Xj, conectados por uma aresta. Ora existe já um caminho de comprimento par entre Xi e Xj, já que existem caminhos, ambos de comprimento par (ou ímpar, no caso de Xi e Xj pertencerem a V2), entre Xi e X e entre X e Xj. Se a esse caminho juntarmos a aresta {Xi;Xj} obtemos um circuito de comprimento ímpar o que contraria a hipótese de apenas existirem circuitos de comprimento par.
- **Grafo bipartido completo** é o grafo bipartido, cujo qualquer vértice do primeiro conjunto é adjacente a todos vértices do segundo conjunto
- **Grafo k-partido** ou **grafo de *k*-coloração** é um grafo cujos vértices podem ser particionados em k conjuntos disjuntos, nos quais não há arestas entre vértices de um mesmo conjunto. Um grafo 2-partido é o mesmo que grafo bipartido.
- **Emparelhamento de grafos** consiste em partir o grafo em conjuntos de vértices a qual não compartilham nenhuma aresta entre eles.
- **Teorema das quatro cores** é baseado no problema das cores necessárias para se colorir um mapa sem que os países vizinhos compartilhem da mesma cor. Transformando o mapa em um grafo pode-se provar que pode-se representar qualquer mapa (um grafo planar) com apenas 4 cores (4 partições).

Buscas em Árvores

Percurso em árvores

- - É possível percorrer de modo sistemático todos os vértices e arestas do grafo. O grafo pode ser dirigido ou não.
  - O percurso em árvores é o processo de visitar cada nó da árvore exatamente uma vez.
  - O percurso pode ser interpretado como colocar todos os nós em uma linha, não existe uma ordem para ser seguida.
  - Existem *n* percursos diferentes, quase todos caóticos.
  - Os básicos são percurso em profundidade e percurso em largura
  - Fila: busca em largura
  - Pilha: busca em profundidade
- **Busca em extensão ou largura**: (Breadth-First Search ou BFS). A propriedade especial está no fato de a árvore não possuir ciclos: dados dois vértices quaisquer, existe exatamente 1 caminho entre eles. Um percurso em extensão é visitar cada nó começando do menor nível e move-se para os níveis mais altos nível após nível, visitando cada nó da esquerda para a direita. Sua implementação é direta quando uma fila é utilizada. Depois que um nó é visitado, seus filhos, se houver algum, são colocados no final da fila e o nó no início da fila é visitado. Assim, os nós do nível n+1 serão visitados somente depois de ter visitados todos os nós do nível n. Computa a menor distância para todos os vértices alcançáveis. O sub-grafo contendo os caminhos percorridos é chamado de breadth-first tree.
- **Busca em profundidade** (Depth-first search ou DFS). Um algoritmo de busca em profundidade realiza uma busca não-informada que progride através da expansão do primeiro nó filho da árvore de busca, e se aprofunda cada vez mais, até que o alvo da busca seja encontrado ou até que ele se depare com um nó que não possui filhos (nó folha). Então a busca retrocede (backtrack) e começa no próximo nó. Numa implementação não-recursiva, todos os nós expandidos recentemente são adicionados a uma pilha, para realizar a exploração. A complexidade espacial de um algoritmo de busca em profundidade é muito menor que a de um algoritmo de busca em largura. A complexidade temporal de ambos algoritmos são proporcionais ao número de vértices somados ao número de arestas dos grafos aos quais eles atravessam. Quando ocorrem buscas em grafos muito grandes, que não podem ser armazenadas completamente na memória, a busca em profundidade não termina, em casos onde o comprimento de um caminho numa árvore de busca é infinito. O simples artifício de " lembrar quais nós já foram visitados " não funciona, porque pode não haver memória suficiente. Isso pode ser resolvido estabelecendo-se um limite de aumento na profundidade da árvore.

Problemas que envolvem grafos

- Coloração de grafos:
  - o Teorema das quatro cores
- Conjuntos de Grafos
  - Conjunto independente
  - Clique
- Problema do Caminho Mínimo
- Árvore de extensão mínima
- Problemas de roteamento:
  - Sete pontes de Königsberg

  - Problema da inspeção de rotas (também conhecido como o "Problema do carteiro chinês")
  - Problema do caixeiro viajante
- Fluxos de rede:
  - Problema de fluxo máximo

Algoritmos importantes

- algoritmo de Dijkstra
- algoritmo de Kruskal
- algoritmo do vizinho mais próximo
- algoritmo de Prim.

Generalizações

Num hipergrafo uma aresta pode conectar mais que dois vértices.

Um grafo não-direcionado pode ser visto como um complexo simplicial consistindo de símplices de uma dimensão (as arestas) e símplices de dimensão zero (os vértices). Ou seja, complexos são generalizações de grafos que permitem símplices de maiores dimensões.

Referências

1. Biggs, N.; Lloyd, E. and Wilson, R. (1986), *Graph Theory, 1736-1936*, Oxford University Press
2. Cayley, A. (1857), «On the theory of the analytical forms called trees», *Philosophical Magazine*, Series IV, **13** (85): 172–176, doi:10.1017/CBO9780511703690.046 (https://dx.doi.org/10.1017%2FCBO9780511703690.046)
3. Tutte, W.T. (2001), *Graph Theory* (https://books.google.com/books?id=uTGhooU37h4C&pg=PA30), Cambridge University Press, p. 30, ISBN 978-0-521-79489-3, consultado em 14 de março de 2016
4. «The Four Color Theorem» (http://www.mathpages.com/home/kmath266/kmath266.htm). Mathpages.com. Consultado em 03 de junho de 2016 Verifique data em: `|acessodata=` (ajuda)
5. Di Battista, Giuseppe; Eades, Peter; Tamassia, Roberto; Tollis, Ioannis G. (1994), «Algorithms for Drawing Graphs: an Annotated Bibliography», *Computational Geometry: Theory and Applications*, **4**: 235–282, doi:10.1016/0925-7721(94)00014-x (https://dx.doi.org/10.1016%2F0925-7721%2894%2900014-x)
6. Longabaugh, William (2012), «Combing the hairball with BioFabric: a new approach for visualization of large networks» (http://www.biomedcentral.com/content/pdf/1471-2105-13-275.pdf) (PDF), *BMC Bioinformatics*, **13**, doi:10.1186/1471-2105-13-275 (https://dx.doi.org/10.1186%2F1471-2105-13-275), PMID 23102059 (https://www.ncbi.nlm.nih.gov/pubmed/23102059)
7. Goodrich, Michael T.; Tamassia, Roberto (2001). *Estruturas de Dados e Algoritmos em Java*. Col: 2ª ed. Porto Alegre: Bookman. p. 502-503. ISBN 85-363-0043-4
8. Goodrich, Michael T.; Tamassia, Roberto (2004). *Projeto de Algoritmos*. Fundamentos, Análise e Exemplos da Internet. Col: . Porto Alegre: Bookman. p. 299-303. ISBN

85-363-0303-4
9. West, Douglas Brent (2001). «Chapter 1 - Fundamental Concepts». *Introduction to Graph Theory*. Col: 2nd ed. USA: Prentice Hall. p. 20. ISBN 0-13-014400-2

Ver também

- Estrutura de dados árvore ordenada - DAGs, árvores binárias e outras formas especiais de grafos.
- Rede complexa
- Teoria espectral de grafos
- Redes de pequeno mundo
- Modelo de Watts e Strogatz

Ligações externas

- Uma Introdução Sucinta à Teoria dos Grafos (http://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf)
- Graph theory tutorial (http://www.utm.edu/departments/math/graph/) (em inglês)
- Graph theory Algorithms (http://students.ceid.upatras.gr/~papagel/project/contents.htm) (em inglês) — algoritmos em pseudocódigo
- Algorithm Animations (http://www2.hig.no/~algmet/animate.html) (em inglês)
- Graph Theory Software (http://graphtheorysoftware.com/) (em inglês)
- Lista de algoritmos de grafos (http://martinbroadhurst.com/Graph-algorithms.html) (em inglês) — com referências e links para bibliotecas de implementação de grafos

Ferramentas de grafos populares

- Graph Visualization Software (http://www.graphviz.org/) (em inglês) — ferramenta para visualizar e interagir com diagramas de grafos
- Visualization of Compiler Graphs (http://www.rw.cdl.uni-saarland.de/users/sander/html/gsvcg1.html) (em inglês)
- Data Visualization Software (http://www.tulip-software.org) (em inglês)
- Planarity Game (http://planarity.net) (em inglês) — jogo sobre Grafos Planares

Obtida de "https://pt.wikipedia.org/w/index.php?title=Teoria_dos_grafos&oldid=48069138"

Categorias: Teoria dos grafos | Pesquisa operacional

- Esta página foi modificada pela última vez à(s) 18h31min de 20 de fevereiro de 2017.

- Este texto é disponibilizado nos termos da licença Creative Commons - Atribuição - Compartilha Igual 3.0 Não Adaptada (CC BY-SA 3.0); pode estar sujeito a condições adicionais. Para mais detalhes, consulte as condições de uso.